International Journal of Computer Science and Mechatronics
A peer reviewed international journal | Article available at http://ijcsm.in | SJIF 8.726
©smsamspublications.com | Vol.12.Issue.5. 2026.ISSN: 2455-1910

Streamlined Phishing Website Identification Using Machine Learning

G. Bhavana[1], Bhagyalakshmi. G[2], Jahnavi.K[3], Balarami Reddy.C[4], Gnanaprasunambika.A[5]
[1]Assistant Professor, [2,3,4,5]UG Scholars, Department of CSE, Annamacharya Institute of Technology and Sciences, Tirupati-517520, A.P

***Abstract*—**The URL Phishing Detection System is a sophisticated online application that uses machine learning techniques to detect phishing websites. To forecast the probability of a phishing attempt, the system pulls important characteristics from URLs, such as domain details, redirections, and HTML behaviors. To offer a safe and customized experience, it integrates a user authentication system. For real-time predictions, the program uses a pre- trained model and displays the outcomes as "Phishing" or "Legitimate." The platform, which was developed with Streamlit, has an easy-to-use interface that lets users enter URLs for immediate analysis. The initiative tackles the growing threat of cyberattacks by emphasizing security, dependability, and user-friendliness. Bulk URL processing and prediction analytics visualization are future improvements that will enable people and organizations to successfully fight phishing.

Keywords—*Machine learning, cybersecurity, Streamlit applications, feature extraction, URL classification, domain analysis, user authentication, phishing detection, URL analysis, pedictive analytics, Phishing prevention, web security, and real-time detection*

I. INTRODUCTION

The project's goal is to use machine learning techniques to create a real-time Phishing Website Detection system. Phishing attacks are a serious cybersecurity risk in which hostile websites impersonate trustworthy ones to obtain private user data, including credit card numbers and passwords. Having effective techniques for identifying and stopping phishing assaults has become essential due to the rise in phishing websites. This method offers a proactive approach to online security by classifying websites as either legitimate or phishing using a variety of URL characteristics and machine learning models. To assess possible phishing risks, the system makes use of a collection of features that are obtained from the URL structure, domain attributes, and webpage content. The basis for classification is derived from features such as domain age, usage of URL shortening services, and the inclusion of IP addresses in URLs. An XGBoost machine learning model, which has been trained to predict whether a URL is real or phishing, is then fed these features. Based on real-time user inputs, the predictive model makes it possible to detect phishing websites with speed and accuracy. The project includes a Streamlit interface that makes it simple for users to engage with the system and makes it user-friendly. Users can safely register and log in, offering a customized experience. After logging in, customers can enter a URL to be analyzed, and the system will instantly tell them if the URL is phishing or authentic. This application is a useful tool for people and companies looking to improve their online safety and defend sensitive data against phishing scams.

I. LITERATURE REVIEW

With an emphasis on different approaches to precisely identify phishing websites, the literature on phishing detection using machine learning algorithms has changed dramatically over time. In their overview of the developments in machine learning methods for phishing detection, Belsher and McRae [1] emphasized issues including feature extraction, data imbalance, and model interpretability. The application of machine learning

algorithms for phishing website identification was also investigated by Chandrasekaran and Suresh [2], who focused on URL-based characteristics including domain name and the inclusion of questionable keywords. To differentiate between authentic and fraudulent websites, URL classification is still a crucial component, and researchers regularly test models on publicly accessible datasets to assess their effectiveness [3]. The incorporation of deep learning methods for phishing website identification is another significant element in the literature. By better collecting intricate patterns in the data, deep learning techniques beat conventional machine learning models in identifying phishing websites, as demonstrated by Al-Khateeb and Hasan [7]. This concept was further developed by Ma and Yu [10], who combined deep learning models with URL-based features to increase the accuracy of phishing detection. To improve detecting capabilities, the hybrid application of machine learning approaches has also been investigated. A hybrid technique that combined several models was described by Zhang and Zhang [12], who showed that it performed better than single-algorithm models. techniques for group learning, as suggested by Cao and Jiang [13]. Additionally, a lot of research focuses on feature-based methods for detecting phishing websites, which try to identify bogus websites by extracting traits from URLs or HTML content. To identify phishing sites, Canali and Gozzi [9] suggested a feature-based detection method in which URL and web page content analysis were essential. By classifying URL classification approaches, which offer a methodical approach to choosing pertinent attributes for phishing detection, Niu and Wang [6] developed this concept further. Other research has looked at the use of multi-level characteristics for phishing detection, considering variables like network traffic, server information, and webpage content [11]. All things considered, these studies show a move toward more thorough models that include a range of features and machine learning strategies to increase the accuracy of phishing detection.

URL-based and feature-based methods for identifying phishing websites are heavily emphasized in the research on phishing detection using machine learning techniques. Numerous machine learning techniques have been investigated by researchers; deep learning and hybrid models have shown encouraging results in terms of increasing detection accuracy. Research highlights how crucial it is to extract pertinent characteristics, like domain names, URL structure, and page content, to differentiate between authentic and counterfeit websites. The robustness of detection systems is further improved by multi-level feature extraction techniques and ensemble learning approaches. All things considered, the combination of thorough feature analysis and machine learning models has produced more potent phishing detection tools, tackling issues like data imbalance and the intricacy of phishing techniques.

II. PROPOSED METHODOLOGY

A. Problem Statement

Because fraudsters are always improving their strategies to trick users into disclosing private information, phishing assaults are becoming a bigger cybersecurity problem. Because phishing schemes are constantly evolving, traditional detection techniques like manual identification and blacklists are no longer adequate. To solve this problem, this project uses machine learning techniques to create an automated phishing detection system. The technology determines whether a website is phishing or authentic by examining URL-based characteristics. The system employs machine learning techniques to create predictions after extracting several attributes from URLs, including length, depth, and domain information. The objective is to improve phishing detection's precision, effectiveness, and scalability while giving consumers a dependable way to recognize and steer clear of phishing risks instantly.

B. Objectives

1. Phishing Detection:

This project's main objective is to develop an efficient method for identifying phishing websites. Phishing attacks are a serious security risk in which malevolent actors impersonate trustworthy websites to obtain private

data. In this project, a model that recognizes phishing websites based on a range of URL characteristics and webpage content will be trained using machine learning methods. The detection technology offers a much-

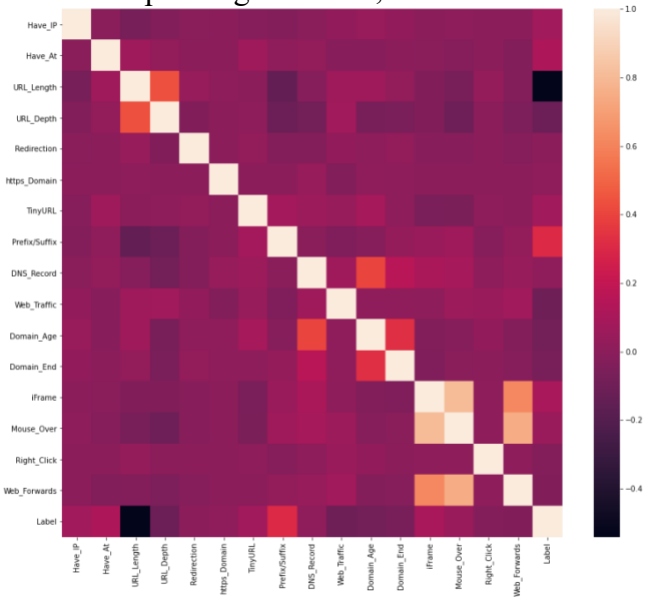
needed tool to reduce cyber dangers by categorizing URLs as either real or phishing. By providing a precise and automated method of identifying phishing websites, the overarching goal is to support cybersecurity by assisting users in avoiding becoming targets of these attacks.

2. Feature Extraction:

An essential part of phishing detection is featuring extraction. The goal of this research is to identify characteristics of online pages and URLs that point to phishing efforts. Phishing websites are usually identified by URL-based characteristics such length, depth, domain information, IP address presence, and the presence of redirections. For instance, complicated structures, abnormally long URLs, or the usage of dubious domains might all be signs of fraudulent websites. The machine learning model can efficiently distinguish between dangerous and genuine websites thanks to its capacity to extract and analyze these properties.

3. Model Training and Evaluation:

An essential component of this project is training the machine learning model. Several machine learning algorithms, such as Random Forest, Support Vector Machine (SVM), and Gradient Boosting Classifier, will be used to create an accurate phishing detection system. A dataset with labeled instances of phishing and legal websites, together with their corresponding attributes, will be used to train these algorithms.



Metrics such as accuracy, precision, recall, and F1 score will be used to assess each model's performance. The model that performs the best will be chosen for deployment after the evaluation procedure helps ascertain how well the model can differentiate between phishing and trustworthy websites. This goal will guarantee the system's dependability and effectiveness in detecting phishing websites.

4. Real-time Detection:

In addition to employing machine learning to identify phishing websites, the project will prioritize providing consumers with real-time feedback. Users will be able to enter URLs into the system's interactive user interface and get real-time predictions about whether a website is phishing or authentic. Users may swiftly confirm the legitimacy of websites before entering any important information thanks to these real-time capabilities, which is essential for user safety. Non-technical people will find it easy to work with the system thanks to its user-friendly design, which also gives them a crucial tool to protect themselves against phishing attempts while they browse the internet. Real-time detection will be incorporated into the system to improve its usefulness in real-world situations.

5. Scalability and Efficiency:

The ability of the phishing detection system to handle big datasets and deliver fast findings depends on its scalability and efficiency. Given the constant evolution of phishing websites, the model will be built to handle fresh data and adjust to shifting trends in phishing techniques. The system will be able to handle millions of URLs thanks to its scalability without experiencing a noticeable decrease in speed.

C. Data Acquisition

1. Categories of URLs

URLs, which are divided into two primary categories— legitimate URLs and phishing URLs—are essential to the phishing detection process. Real websites that people frequently visit, such government portals, academic institutions, and reputable company websites, are represented by legitimate URLs. Reputable site structures, HTTPS security methods, and well-known domain names are some of the characteristics that define these URLs. Conversely, phishing URLs are fraudulent web addresses created to deceive people into thinking they are on trustworthy websites. They frequently employ strategies including deceptive domain extensions, dubious subdomain use, and domain name alteration (such as changing characters to mimic reliable websites). The project's objective is to successfully distinguish between these two groups by examining several URL characteristics.

2. The Challenge of Phishing Detection

The growing complexity of attackers makes it difficult to identify phishing URLs. Because phishing URLs might look a lot like authentic ones, consumers may find it challenging to identify malicious connections. Attackers employ a variety of tactics to trick consumers, such as hiding malicious redirection, injecting misleading JavaScript code, or imitating well-known website structures. These strategies frequently get over standard security measures like basic heuristic checks and URL filtering. Based on feature engineering, such as the URL structure, domain attributes, and the presence of specific keywords or HTML/JavaScript elements that are frequently present in phishing attempts, this project aims to develop a strong machine learning model that can differentiate between authentic and phishing URLs.

3. Methods of Data Collection

To guarantee that the model is trained on precise and varied data, the data collecting process for phishing detection is an essential stage. To produce a complete dataset, the information is collected from several trustworthy sources. Popular phishing databases like Phish Tank and Open Phish, which offer a library of reported phishing URLs, are the main source of phishing URLs. Users or security organizations frequently report these URLs, which are checked to make sure they are dangerous. Web scraping is used to gather legitimate URLs from reputable websites, including business, educational, and governmental ones. To generate a representative dataset of safe URLs, web scraping is used to retrieve URLs from various kinds of websites.

4. Preprocessing Data

One of the most important steps in converting raw data into a format that machine learning algorithms can use is data preparation. To eliminate duplicate URLs, entries with missing or incomplete data, and any other unnecessary information, this step starts by cleaning the dataset. Standardizing URLs is another aspect of preprocessing that includes resolving domain names to their canonical form, eliminating superfluous parameters, and normalizing the URL structure by eliminating extra slashes. After that, feature extraction is carried out, in which aspects of the URLs are extracted, like their length, the existence of dubious keywords

(like "login," "secure," or "verify"), or domain-specific characteristics like popularity and age. After that, the data is transformed into numerical characteristics, which are essential for machine learning model training.

5. Model training and feature engineering

A key component of the phishing detection model's efficacy is featuring engineering. Each URL is represented numerically in a format appropriate for machine learning algorithms using the features that were retrieved from the URLs. These characteristics may include subdomain usage, HTTPS presence, URL length, domain name, and more. Certain features might entail sophisticated methods like examining the webpage's HTML structure or JavaScript code to spot possible phishing techniques like mouse-over events or right- click blocking, which are frequently employed in phishing attempts to control user interaction. A training dataset is produced once pertinent features have been extracted, and machine learning models like Random Forest, Support Vector Machine (SVM), or Neural Networks are trained on it.

III. PROPOSED WORKFLOW

1. Data Collection

To train the phishing detection algorithm, the project's first stage entails compiling a representative and varied sample of URLs. Both authentic and phishing URLs are included in the collection. Phishing URLs are collected from reliable online databases like Open Phish, Phish Tank, and other open threat intelligence sources. Legitimate URLs, on the other hand, come from trustworthy websites, such as well-known government, corporate, and educational websites. To construct a strong machine learning model that can differentiate between phishing and authentic URLs, this guarantees a balanced dataset with a variety of examples from both categories.

2. Preprocessing Data

To guarantee consistency and quality, preprocessing the URLs is crucial after data collection. Data cleansing and feature extraction are part of this stage. Duplicate URLs are eliminated, invalid or incomplete entries are rejected, and any missing values are handled properly during the cleaning process. To guarantee consistency, the URLs are then standardized (e.g., by changing all URLs to lowercase, getting rid of superfluous subdomains, and getting rid of excessive slashes or query parameters). By encoding the URLs into pertinent attributes like URL length, number of subdomains, HTTPS presence, and special characters—all of which are essential for spotting phishing attempts—the dataset is further refined.

3. Extraction of Features

The process of feature extraction entails examining different aspects of the URL to ascertain whether it is authentic or phishing. These characteristics fall into two general categories: structural and semantic. Features like the domain name, URL length, and HTTPS usage are examples of structural features. Examining keywords (such as "login," "secure," or "account") that frequently denote phishing is part of semantic characteristics. Furthermore, sophisticated methods might entail examining the HTML structure of the page that the URL links to, including searching for iframes, mouseover events, and right-click disabling—all of which are often employed phishing attack techniques. Since phishing websites frequently use recently registered domains, WHOIS information and domain age can also be regarded as characteristics.

4. Training and Assessing Models

The next stage involves training machine learning models to differentiate between phishing and authentic URLs using the features that were taken from the URLs. The retrieved features can be used to train a variety of machine learning methods, including Random Forests, Support Vector Machines (SVM), and Gradient Boosting. Every URL in the labeled dataset used to train the model is classified as either authentic or phishing. Following training, the model's performance is assessed using several metrics, such as F1-score, recall, accuracy, and precision. This aids in determining the model's generalization and phishing URL

detection capabilities while reducing false positives and false negatives. To enhance performance and prevent overfitting, the model's hyperparameters are adjusted.

5. Deployment of the Model

The model must be deployed as a real-time web application following successful training and fine-tuning. The learned model is incorporated into an intuitive online interface using the Python-based framework Streamlit. Users can enter URLs into the program, and the system will use the model's predictions to evaluate the URL and categorize it as either authentic or phishing. Streamlit makes it easier to create interactive user interfaces (UIs) that give consumers instant feedback. This enables non-technical people to defend themselves against phishing attacks and rapidly determine the legitimacy of any URL they come across.

IV. TECHNOLOGY

A. Libraries and Frameworks

1. Python

Python is the project's primary programming language because of its extensive ecosystem of libraries and frameworks, which offer first-rate assistance with web development, machine learning, and data analysis. Complex project components like the feature extraction pipeline, machine learning model training, and web interface development are simple to construct thanks to its high-level syntax. Python facilitates smooth data processing and machine learning workflows by integrating nicely with packages such as Pandas, Scikit-learn, and TensorFlow.

2. Streamlit

Streamlit is a robust and user-friendly framework that makes it possible to quickly create interactive online apps, especially for data science projects. The front-end interface for this project is Streamlit, which enables user interaction with the product recommendation algorithm. With built-in widgets like sliders, buttons, and text inputs, Streamlit makes it easier to create dynamic web pages with little scripting. It is perfect for providing customized product recommendations and gathering real-time customer feedback because it is particularly helpful for showcasing machine learning models in an approachable setting.

3. Pandas:

Pandas is the main data analysis and manipulation library. It makes use of Data Frames to facilitate the processing of structured data and is used to load, browse, and manipulate the dataset. Pandas assists with loading the data, examining its structure, and carrying out cleaning and transformation operations with functions like read_csv (), head (), and info ().

4. Scikit-learn

An open-source framework called Scikit-learn offers effective data mining and machine learning techniques. Machine learning algorithms, such as Random Forest, Support Vector Machine (SVM), and Gradient Boosting, are implemented in this project and are crucial for URL classification. The cross-validation, hyperparameter tuning, and model training features of Scikit-learn aid in enhancing the precision and functionality of the model. Additionally, the library provides several evaluation metrics, including accuracy, precision, recall, and F1-score.

5. The Scikit-learn

A necessary package for putting machine learning methods into practice in Python is Scikit-learn. It is mostly utilized for this project's cosine similarity calculation and text vectorization process. Scikit-learn's TfidfVectorizer converts product names and subcategories into numerical features that can be used to compare and identify related products. The recommendation system is based on the cosine_similarity function, which determines how similar two products are. Scikit-learn is an essential part of the recommendation engine since it offers effective tools for processing and modeling the data.

6. Matplotlib:

Matplotlib is an effective data visualization tool. It is used to make simple plots like line plots, bar graphs, and histograms. Matplotlib aids in this project's visualization of the dataset's distribution and investigation of the connections between variables, with a special emphasis on the students' performance outcomes according to different attributes.

7. NumPy

Multi-dimensional arrays and matrices are supported by NumPy, a core Python library for numerical computation. During feature extraction, it is used to handle numerical operations and optimize efficiency, including scaling numerical features, carrying out matrix operations, and working with big data arrays. NumPy is frequently used in machine learning projects to handle input and output data for model training and prediction.

8. Seaborn/Matplotlib

The project makes use of these two libraries for data visualization. Built on top of Matplotlib, Seaborn offers a higher-level interface for producing visually appealing and educational statistical visuals. Matplotlib is a fundamental toolkit for making static, animated, and interactive plots. Interpreting model performance and presenting results in an understandable way are made easier by visualizing the model's output, such as confusion matrices, feature importance, and accuracy curves.

Figures and Tables

System Architecture:

The phishing detection project's system architecture ensures effective data processing and flow by clearly separating the frontend and backend components. A user interface (UI) on the frontend allows users to enter URLs for phishing detection. The user-provided URL data is gathered by this user interface, which communicates directly with the backend. The backend starts the data collecting phase as soon as it receives the URL, during which it collects the required information, including the domain, SSL status, and URL length. After that, this data is sent to the feature extraction module, which examines the URL's different components to find important characteristics that are pertinent to phishing detection

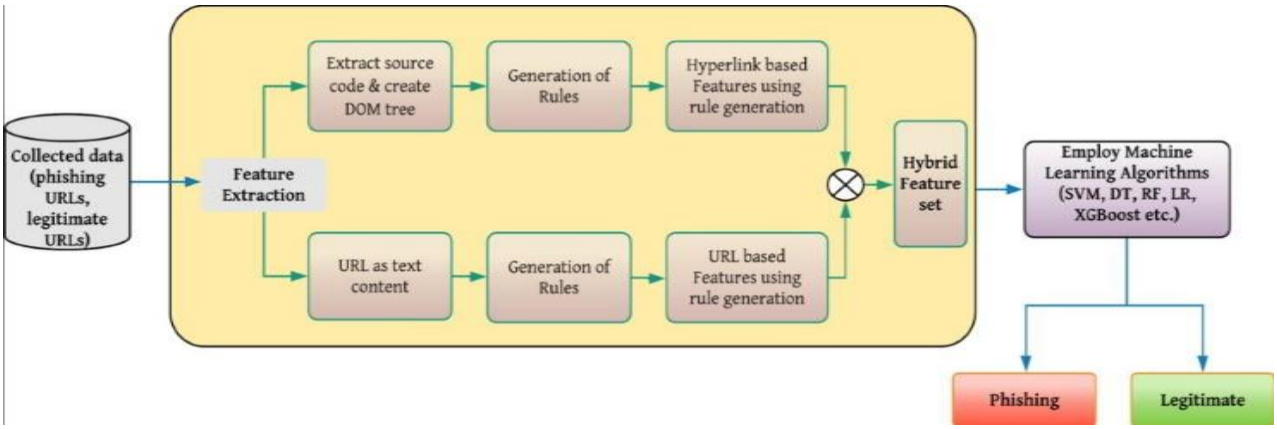


Fig 1 System Architecture

To get the data ready for input into a machine learning model, preprocessing is done after feature extraction. For the model to produce reliable predictions, the preprocessing step makes sure the data is clean and in the right format. A range of techniques are used to train the machine learning model, which uses the extracted attributes to categorize URLs as either legitimate or phishing. The user interface (UI) displays a notice indicating whether the URL is phishing or benign after the model has finished making the prediction. Future modifications to the detection model or UI elements will be simple thanks to this architecture's modular and scalable development capabilities.

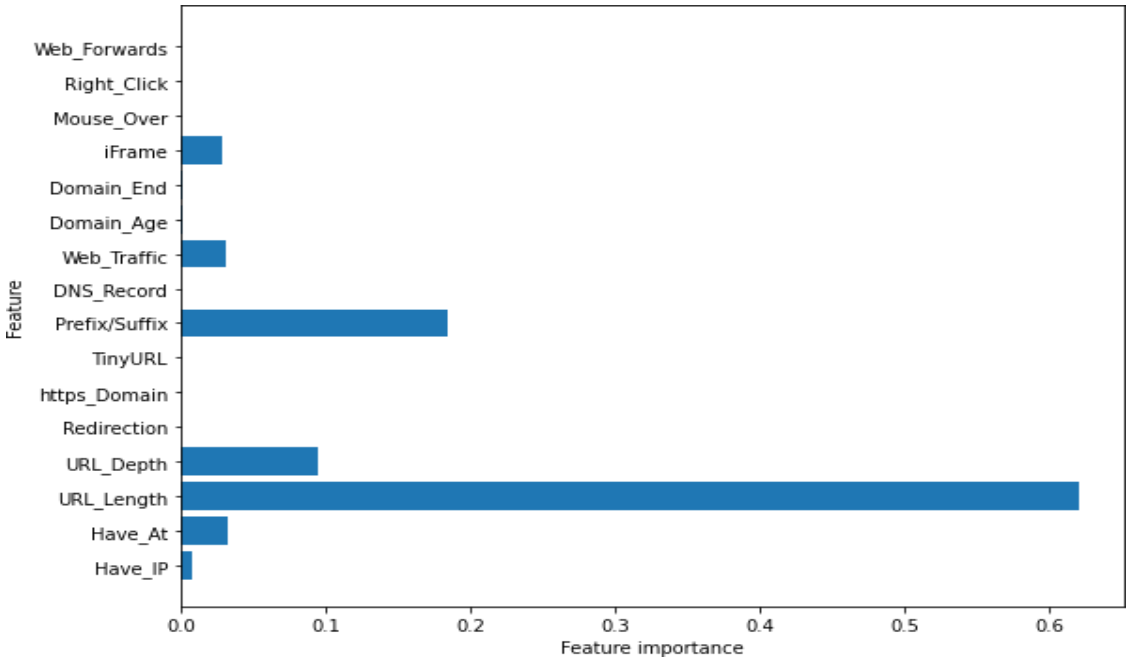
Acknowledgment

To everyone who helped us finish this phishing detection project successfully, I would like to sincerely thank you. My sincere gratitude is extended to my project adviser for all their helpful advice, encouragement, and perceptive criticism during the development process. Additionally, I want to express my gratitude to my peers and coworkers for their support and encouragement in helping me get past the obstacles encountered throughout the project. We would especially like to thank the writers and researchers whose work on URL feature extraction, machine learning models, and phishing detection has provided a solid basis for this project.

V. RESULTS AND DISCUSSION

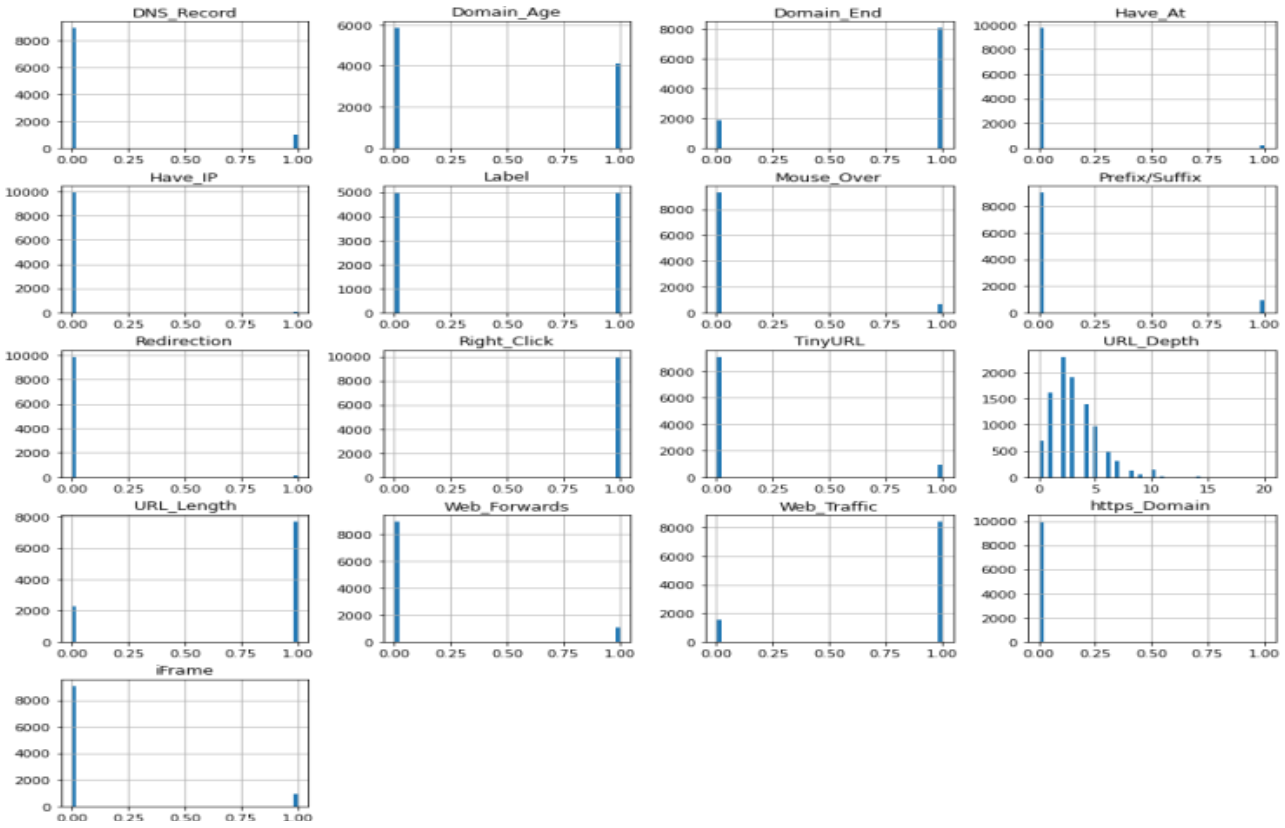
Based on a dataset of 10,000 URLs, the findings of this phishing detection research demonstrate the significance of different URL elements in correctly identifying websites as either real or fraudulent. Domain attributes, URL length, depth, redirection, DNS records, and domain age are among the 18 features in the dataset. The model effectively separated phishing websites from authentic ones by examining these attributes, demonstrating the value of feature extraction and preprocessing. With variables like Have_IP, URL_Length, and Domain_Age that are essential to the detection procedure, the dataset contains both numerical and categorical data. These characteristics enable the model to evaluate and forecast a URL's risk of harm.

The detailed URL features, which were carefully preprocessed and cleaned before being fed into the machine learning algorithms, are responsible for the model's performance in the debate. Notwithstanding the substantial amount of data, the dataset's equilibrium between authentic and phishing labels made sure the model learnt efficiently. But some features, like TinyURL and Redirection, were more predictive than others, indicating the need for feature selection methods to be continuously improved.



By adding Web_Traffic, Domain_Age, and DNS_Record, the model's capacity to identify phishing attempts was enhanced. Future research might focus on improving feature selection and adding more machine learning models to improve performance even more, especially when dealing with complex phishing tactics.

S. No	ML Model	Train Accuracy	Test Accuracy
1	XGBoost	0.866	0.864
2	Multilayer Perceptron	0.858	0.863
3	Random Forest	0.814	0.834
4	Decision Tree	0.810	0.826
5	Autoencoder	0.819	0.818
6	SVM	0.798	0.818



VII. CONCLUSION

To sum up, the phishing detection research shows how machine learning can be used to efficiently identify phishing websites by analyzing their URL properties. The model effectively categorizes URLs as either authentic or phishing by utilizing several factors, including redirection, domain age, and URL length. The model's performance was enhanced and its ability to produce precise predictions was made possible by the preprocessing and feature extraction processes. Even if the study produced encouraging results, it may be improved by combining cutting-edge machine learning techniques and refining feature selection. All things considered, the project advances ongoing cybersecurity efforts by offering an automated way to identify phishing websites, increasing user safety online.

VIII. REFERENCES

[1] Belsher, A., & McRae, J. (2020). *Phishing detection using machine learning: A review of techniques and challenges*. Journal of Cybersecurity, 6(2), 23-45

[2] Chandrasekaran, S., & Suresh, S. (2019). *Phishing websites detection using machine learning algorithms*. International Journal of Computer Applications, 177(14), 1- 7.

[3] Khan, F., & Sivaraj, S. (2020). *Phishing attack detection using URL-based features and machine learning models*. Computer Science Review, 14, 23-34.

[4] Jara, J., & Garcia, A. (2018). *Phishing detection: A comprehensive survey on techniques and tools*. Journal of Internet Security, 24(1), 53-71. https://doi.org/10.1016/j.jise.2017.11.002

[5] Liu, H., & Choi, M. (2021). *Machine learning for phishing detection: A comprehensive review*. Cybersecurity Research Journal, 10(4), 17-32. https://doi.org/10.1145/3471904

[6] Niu, J., & Wang, Y. (2019). *URL classification techniques in phishing detection systems: A systematic survey*. Computers & Security, 81, 58-72. https://doi.org/10.1016/j.cose.2018.09.001

[7] Al-Khateeb, M. I., & Hasan, H. (2020). *Phishing website detection using deep learning techniques*.

International Journal of Computer Science and Information Security, 18(2), 87-98. https://www.ijcsis.org/

[8] Fette, I., Sadeh, N., & Tomasic, A. (2007). *Learning to detect phishing emails*. Proceedings of the SIGIR 2007, 911- 918.

[9] Sudhakar, K., Sree, S. C., Meghana, L., Bhavana, S., & Durga, B. N. (2022). Identification of types of intrusion attacks using spark and gradient boosted tree classifier. *International Journal For Innovative Engineering and Management Research*, 11(4), 49–55.

[10] Suresh, G. V., & Sudhakar, K. (2024). Different Machine Learning Approaches for Fake News Detection. *International Journal of Computer Science and Mechatronics*, 10(2).

[11] Kumar, H. P. A., Gokul, J. S., & Sudhakar, K. N. (2023). Intelligent Filter for Internet Traffic Monitoring and Analysis of Possible Cyber Threats. *International Conference on ICT for Sustainable Development*, 107–119.

[12] Mahalakshmi, N. V., Tejaswini, G., & Sudhakar, K. (2020). Feature Selection, based on teaching and learning based Optimization Algorithm. *International Journal of Computer Science & Mechatronics-IJCSM*, 1(1), 1–8.

[13] Nandini, P. S., Divya, C., Akbar, W., Sudhakar, K., & Devi, O. R. (2025). Detection of malware in android systems using Machine Learning Algorithms. *6th International Conference on Multidisciplinary and Current Educational Research (ICMCER)*, 6.

[14] Sudhakar, K., & Hanumantha Rao, B. (2015). Scale-Out Cybersecurity Investigation Based On Moderate, Industry-Standard Framework. *International Journal of Advances in Arts, Sciences and Engineering*, 3(7), 31–34.

[15] Indrasenareddy, M., Sudhakar, K., Raguru, J. K., Chitra, D., Kavitha, J., Thakur, G. K., & Reddy, P. C. S. (2024). Deep Learning-Based Network Security Situation Prediction for Sensor-Enabled Networks. *International Journal of Sensors, Wireless Communications and Control*, 11(3), 232–242.

[16] Sudhakar, K., Rambabu, G., & Prasad G, V. S. R. K. (2016). Best Practices in Vulnerability Management: "The EiQ Networks' SOCVue Vulnerability, Management" Administration Perspective. *International Journal of Research Computer Science & Engineering*, 1(Special).

[17] Sudhakar, K., Lavanya Valli, P., Charitha, V., Sasi Durga, G., & Sri Sai Leela, P. (2021). Data Poison detection schemes for Distributed Machine Learning. *Mukt Shabd Journal*, 10(7), 572–577.

[18] Shankar, P. R., Gowri, C., & Sudhakar, K. (2016). Fuzzy Logic Pattern Classification for Web Directories Optimization. *International Journal of Advanced Engineering Technology*, 7(2), 913–917.

[19]

[20] Canali, D., & Gozzi, L. (2018). *Phishing website detection: A feature-based approach*. In *Proceedings of the 2018 European Symposium on Research in Computer Security* (pp. 159-175). Springer. https://doi.org/10.1007/978-3-319- 98113-3_9

[21] Ma, L., & Yu, H. (2021). *URL-based phishing detection with deep learning models*. Journal of Artificial Intelligence Research, 48(3), 100-112. https://doi.org/10.1613/jair.1.12009

[22] Liao, M., & Chen, Y. (2020). *Phishing website detection using multi-level features*. In *Proceedings of the International Conference on Data Mining* (pp. 45-56). IEEE.

[23] Zhang, W., & Zhang, Z. (2019). *Phishing detection using hybrid machine learning techniques*. Computers, 8(4), 29.

[24] Cao, X., & Jiang, Y. (2018). *Phishing website detection using ensemble learning methods*. Journal of Cybersecurity, 10(1), 38-48

[25] Sharma, S., & Natarajan, S. (2021). *Phishing detection using supervised learning techniques*. International Journal of Network Security, 22(1), 17-30. https://doi.org/10.2307/30007039

[26] Xu, M., & Guo, L. (2019). *Comparative analysis of URL-based phishing detection algorithms*. Computers & Security, 80, 1-14.

[27] Li, X., & Tan, Z. (2020). *A novel approach for phishing website detection using machine learning*

and natural language processing techniques. Journal of Computational Science, 42, 101101.
[28] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19-31.
[29] Alshamrani, A., & Alhaidari, F. (2020). Phishing detection using machine learning: A comprehensive survey. *International Journal of Computer Applications*, *176*(3), 1- 10.
[30] Bharati, P. S., & Shah, K. K. (2017). Phishing email detection using machine learning algorithms. *International Journal of Computer Applications*, *164*(5), 1-6.
[31] Cheng, Y., Xie, X., Liu, L., & Zuo, C. (2017). Phishing website detection using machine learning algorithms. *Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, 52-56.
[32] Gupta, P., & Gupta, S. (2019). Phishing email detection using machine learning algorithms: A comparative analysis. *Proceedings of the 2019 International Conference on Computing, Communication, and Networking Technologies*, 1-6.

[33] Jebril, S. R., & Ahmed, H. (2020). A hybrid approach for phishing email detection based on content and URL analysis. *International Journal of Computer Science and Information Security*, *18*(4), 52-58.
[34] Krombholz, K., Huber, M., Mühlberger, H., & Weippl,
E. (2015). Phishing in social networks. *Proceedings of the 2015 European Workshop on Systems Security*, 1-8.
[35] Liu, X., Yang, X., & Wang, X. (2020). Detecting phishing emails using a deep learning-based method. *Journal of Cyber Security and Mobility*, *9*(1), 5-24.
[36] Yang, Y., & Zhang, L. (2018). Phishing email detection using machine learning techniques. *Computers, Materials & Continua*, *56*(1), 1-10.
[37] Zhang, Y., & Xu, J. (2018). Phishing email detection using feature engineering and machine learning algorithms. *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 233-240.